

Doğru İşlemci Seçimi İçin Uzman Bir Yaklaşım

Berat YILMAZ¹, İsmail ŞAHİN²

¹Bilişim Enstitüsü, Gazi Üniversitesi, Ankara, Türkiye

²Teknoloji Fakültesi, Gazi Üniversitesi, Ankara, Türkiye

brtyilmaz@gmail.com, isahin@gazi.edu.tr

(Geliş/Received:23.01.2017; Kabul/Accepted:20.06.2017)

DOI:10.17671/gazibtd.331040

Özet— Bu çalışmada; güncel bilgisayar işlemcileri arasında tüketicinin ihtiyacını en iyi karşılayan modeli, kullanıcıya yönelttiği sorulara bağlı olarak tespit etmeye çalışan bir dinamik uzman sistem geliştirilmiştir. Bu çalışmanın en belirgin özelliği, oluşturulan kuralların sınırsız uzunlukta ve karmaşıklıkta yaratımına olanak veren bir veri yapısı ve bu veri yapısını işleyebilecek kapasiteye sahip analiz mekanizmasını içeriyor oluşudur. İşlemci dünyasına her geçen gün katılmakta olan 10'larca modelin sübjektif bakış açısından uzak somut bir karşılaştırma ile değerlendirmek için güncel donanımsal karşılaştırma sitelerinin verileri kullanılmıştır. Böylelikle çalışmanın temelindeki dinamiklik yaklaşımı veri içeriği ve çeşitliliği (bilgi tabanı) açısından da sağlanarak komple sistemin bu prensip üzerinde şekillenmesi sağlanmıştır. Çalışma temelinde, uzman tarafından hazırlanan, kullanıcıların performans eğilimini ölçmeye yönelik soruları anket yöntemiyle sunmayı ve alınan cevapları analiz ederek güncel işlemciler arasında bir sonuç alt kümesi yaratıp kullanıcıya karar vermede yardımcı olmayı hedeflemektedir.

Anahtar Kelimeler— işlemci, doğru işlemci seçme, merkezi işlem birimi tercihi, donanımsal karşılaştırma, dinamik uzman sistem

An Expert Approach for Choosing the Appropriate Processor

Abstract— In this study, a dynamic expert system approach has been developed. In this system, the needs of the customer are met in order to identify the model which fits best to the needs of the customer. For this purpose, the questions are directed to the user by considering the contemporary computer processors. The most significant features of this work are a data structure which has unlimited length and complexity, and an analysis mechanism which has the ability to handle this data structure. In our work, the data of up-to-date hardware comparison websites are chosen to evaluate the models which are participating to the world of processors day by day. These data are processed by system to make objective and concrete comparison. By using these data, the dynamism which establish a base for our work additionally demonstrate a base both for data content and data diversity (knowledge base). Consequently, the whole system aims to ask the questions to the user and to analyze the answers given by the user. After that, these answers are processed to build a subset of appropriate processors. Hence, this subset is used to help the user.

Keywords— processor, choose the right processor, central processing unit preference, hardware comparison, dynamic expert system

1. GİRİŞ (INTRODUCTION)

Günümüzde kullanıcıların dizüstü, masaüstü, iş istasyonu gibi doğrudan alacakları ya da özelleştirip toplayarak dolaylı yoldan oluşturacakları sistemlerde işlemci seçimi konusunda zorlandıkları aşikardır [1]. Bu işlemci iyi midir? İnsanlar tarafından tavsiye ediliyor mu? Verdiğim paranın performans anlamında karşılığını alabilir miyim? Daha ucuz bir modele mi yönelsem? Gibi sorular bu süreçte kullanıcıların aklını meşgul eden belli başlı

sorulardandır. Piyasadaki işlemci çeşitliliği ve bu çeşitliliğin her geçen gün artması, hatta aynı işlemcilerin kendi içerisinde birden çok jenerasyona ayrılmasıyla karşımıza ciddi bir seçim karmaşası çıkarmaktadır [2].

Kullanıcının işlemciden ne beklediğini bilmesi kendisini gereksiz harcamalardan, performans noktasında hayal kırıklıklarından, fazla enerji tüketiminden dolayı gelecek ısınma ve yüksek tutarlı elektrik faturalarından alıkoymaz.

İşlemci tercih edilirken ihtiyaçların sorgulanması gerekmektedir. Oyun, çoklu ortam ve mühendislik uygulamaları işlemci seçiminizi değiştirebilecek etkenlerden sadece birkaçıdır. Peki hangi amaç için nasıl bir işlemci gerekiyor? Oyun için çekirdek sayısı mı önemli yoksa ön bellek mi? Saat frekans hangi aralıkta seçilmeli? Oyun için AMD mi yoksa Intel marka işlemci mi seçilmeli? Tüm bu sorulara cevap vermeden önce şu konuyu belirtmekte fayda var. Bugün işlemcilerin performansları tek başına ne saat hızları ne çekirdek sayıları ne de ön bellek miktarları belirliyor. İşlemcilerin performansını belirleyen önemli etkenlerin başında üretiminde kullanılan tasarım mimarisi yer alıyor [3].

Kullanıcıların birçoğunun, işlemci mimarilerini yakından tanıma ve buna bağlı olarak karar alma durumları söz konusu olmadığı için yanlış işlemci tercihlerinde bulunulması işten bile değildir. Söz konusu probleme bir çözüm olarak, eğer mimari alanda yeterli bilgi birikimine sahip kişilerin (uzmanların) bilgilerini bir sisteme aktarıp, metodolojik yollar ile kullanıcıların bu uzman kişilerin bilgilerinden faydalanması sağlanırsa, doğru işlemcinin seçimi konusunda ciddi bir yol kat edilmiş olur [4].

Uzmanın düşüncesini, bünyesindeki sonlu sayıdaki soru ve bu soruların seçenekleri ile ilişkilendirilmiş kurallar ile koda aktarılmasını sağlayan "Cpu Değerlendirme" yazılımı makalenin ilerleyen başlıklarında derinlemesine incelenecektir.

2. UZMAN SİSTEMLER (EXPERT SYSTEMS)

Uzman sistemler için literatürden benzer anlama gelen birçok tanımlama bulunmaktadır. Yapılan çalışmanın daha iyi anlaşılabilmesi açısından, aşağıda bu tanımlamalardan birkaçına yer verilmiştir.

- Uzman sistem (US), bilgi tabanlı bir karar destek sistemidir. Aynı zamanda, daha iyi karar vermeye yardım eden bir araçtır [5].
- US, belirli bir konuda uzman olan bir veya birçok insanın yapabildiği muhakeme ve karar verme işlemlerini modelleyen bir yazılım sistemidir [6,7].
- Bir US programı, belli bir algoritmaya dayanmayan, kendi bilgi tabanı içerisinde, girilen veya önceden belli olan verilere göre arama yaparak bu veriye veya verilere uygun olan bilginin (kuralın) aktifleşmesini sağlayan ve bu aktifleşme sonucu yeni bir veri elde ederek aramaya devam eden bir sistem şeklinde çalışmaktadır [8].
- US'ler, yapay zeka konusunun bir dalı olup, temelde uzman bir insan düzeyinde problem çözmede, insan bilgisini yoğun biçimde kullanan programlardır [9].
- US, özel birtakım problemlerin çözümünde, uzmanların bilgisini ve usavurma sürecini taklit etmeyi amaçlayan, danışman bilgisayar programlarıdır [10].
- US, yüksek performans elde etmek için, uzman bilgileri kullanan bir bilgisayar programıdır [11].

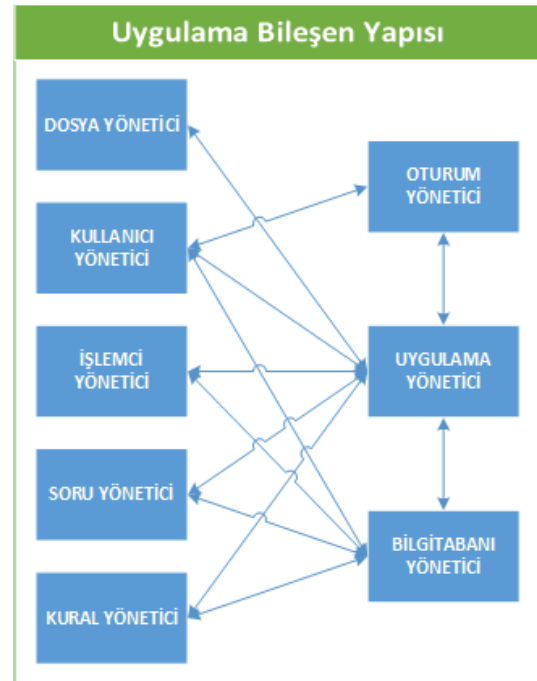
3. CPU DEĞERLENDİRME UYGULAMASI (CPU EVALUATION APPLICATION)

Cpu, İngilizcede merkezi işlem birimi ifadesinin kısaltması olup bilgisayar bilimlerinde (Von Neumann Mimarisi) işlem yapılan merkezi adresler [12]. Cpu Değerlendirme uygulaması genel olarak performans açısından analiz edilmiş işlemci verilerini ve kullanıcıdan aldığı bilgileri kullanarak kullanıcıya doğru işlemciyi alma sürecinde destek olmayı hedefler.

Uygulama, C# programlama dili [13] ile .Net 4.5 platformu kullanılarak, Visual Studio 2015 geliştirme ortamı üzerinde kodlanmıştır. Kaynak kodun kontrolünü sağlamak amacıyla Team Foundation Server 2015 yazılımı tercih edilmiştir [14]. Kullanıcı arayüzleri DevExpress(v15.2) Winform kütüphanesinden faydalanılarak geliştirilmiş olup [15], veritabanı yönetim sistemi olarak da Microsoft SQL Server 2014 kullanılmıştır. Böylece C# ile MsSQL birlikteliğinde kullanılabilen LINQ to SQL tekniğinden faydalanılmıştır [16].

3.1. Bileşen Yapısı (Component Structure)

Cpu Değerlendirme Uygulaması bileşen temelli bir tasarıma sahiptir [17]. Bu tasarımda tüm bileşenlerin çerçeveleri, görev tanımları göz önünde bulundurularak çizilmiş olup, yapacakları işlemler ve birbirleri ile olan etkileşimleri kodlama öncesinde belirlenmiştir. Şekil 1'de bu bileşen yapısı ve bileşenlerin birbiri ile olan etkileşimi göstermektedir.



Şekil 1. Uygulama bileşen yapısı
(Component structure of the application)

3.1.1. Uygulama Yönetici (Application Manager)

Uygulama yönetici bileşeni, adından da anlaşılacağı üzere yazılımın yönetimini sağlayan temel bileşendir. Yazılımın başlatılmasıyla beraber görev alan bu bileşen diğer tüm bileşenleri(servisleri) başlatmakla ve yazılımın kapatılması sürecinde tüm bu servisleri problemsiz bir şekilde sonlandırmakla görevlidir. Ayrıca çok izlekli (multithreaded) [18] bir tasarım anlayışıyla geliştirilen Cpu Değerlendirme Uygulamasında, her bir iş parçacığını yaratan, yöneten ve zaman uyumlamasıyla ilgilenen yine uygulama yönetici bileşenidir.

3.1.2. Oturum Yönetici (Application Manager)

Uygulama, kullanıcılarına farklı oturumlar ile hizmet vermektedir. Bu oturumların yönetiminden sorumlu bileşen ise oturum yönetici bileşenidir. Yazılıma doğru bilgiler ile giriş yapan kullanıcılar için yetki seviyelerine göre yönetici veya kullanıcı oturumlarının açılmasını sağlar. Bu bileşen oturum kapsamında gerekli tüm bilgileri çalışma zamanı süresince ilgili diğer bileşenlere sağlar. Ayrıca bu bileşen geçerli oturumu kapatma, farklı bir oturuma geçiş yapma gibi özellikler sunar.

3.1.3. Bilgi Tabanı Yönetici (Knowledge Base Administrator)

Bileşenin açıklamasına başlamadan önce birkaç ifadeye açıklık kazandırmak yerinde olacaktır.

Veri sınıfları, yazılımın çalışma zamanında ihtiyaç duyduğu verileri tutan sınıflardır. Yazılım kapatılırken veri sınıflarının içeriği veritabanına yazılır; yazılım başlatılırken ise yazılan bu veriler tekrar veritabanından okunarak veri sınıfları içeriğine geri alınır.

Bilgi tabanı, bu kavram uzman sistem özelinde kuralları içeren kural tabanı ve kullanılan verileri içeren veritabanının bütünü için söylenen genel bir adlandırmadır.

Uygulama kalıcı olması beklenen tüm bilgilerini arka planda çalışan veritabanı yönetim sistemi üzerinde tutmaktadır. Bu bileşen ihtiyaç halinde veritabanından okuduğu verileri veri sınıflarına yazar ve aynı şekilde ihtiyaç duyduğunda veri sınıflarındaki verilerin veritabanına kaydedilmesini sağlayarak yazılımın başarıyla işletilmesini sağlar. Özellikle veriyi veritabanına yazma işi maliyetli bir işlem olduğundan bu kısımda özel optimizasyonlar yapılmış ve sadece üzerinde değişiklik yapılmış olan veri sınıfı verilerinin veritabanına yazılması sağlanmıştır.

3.1.4. Dosya Yönetici (File Manager)

Yazılımın ihtiyaç duyduğu yapılandırma ayarlarının XML formatında bir dosyada saklanmasını ve ihtiyaç halinde yine bu dosyadan okunmasını sağlayan bileşendir.

3.1.5. Kullanıcı Yönetici (User Manager)

Uygulamada, kullanıcı işlemlerinin yönetimi üstlenen bu bileşen; kullanıcı ekleme, düzenleme, silme gibi temel işlemleri üstlenmiştir. Ayrıca kullanıcı yetkilendirmelerini düzenler ve oturum yöneticiye gerekli bilgileri sağlar.

3.1.6. İşlemci Yönetici (Processor Manager)

Uygulamada işlemci verileri ile ilgili yönetimi üstlenen bu bileşen; internet üzerinden güncel işlemci verilerini çekme, manuel işlemci ekleme, var olan işlemci verilerini düzenleme ve silme gibi işlemleri üstlenmiştir.

3.1.7. Soru Yönetici (Question Manager)

Uygulamada; anket yönetimi ve bu anket için soru ve seçenekler ekleme, düzenleme, silme gibi işlemleri üstlenen bileşendir.

3.1.8. Kural Yönetici (Rule Manager)

Uygulamada; kural ekleme, düzenleme, silme, listeleme ve görselleştirme gibi işlemleri üstlenen bileşendir.

3.2. Uygulamanın Yapısı ve Özellikleri (The Properties and Structure of the Application)

Cpu Değerlendirme yazılımı kullanıcılarına modern işlemciler arasından kullanım ihtiyaçlarını en iyi karşılayacak modelleri öneren bir uzman sistemdir. Bu öneri süreci şu şekilde çalışır:

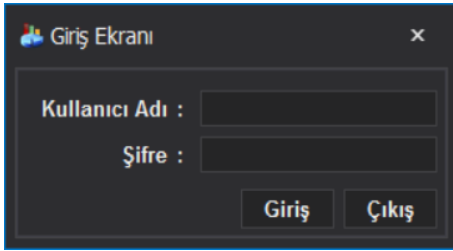
- Sisteme yönetici yetkisi ile giren kullanıcı güncel işlemci verilerini internet üzerinden otomatik olarak sisteme indiren işlevi çalıştırır ve yazılımın işlemci veritabanını güncellemiş olur.
- Sisteme yönetici yetkisine sahip uzman tarafından sorular eklenir. Bu sorular kullanıcının işlemciden beklentilerini ölçmeye yönelik soru ve seçenekler içermektedir.
- Sisteme yönetici yetkisiyle giriş yapan uzman, sisteme eklediği sorular ve bu soruların seçenekleri ile istediği karmaşıklıkta ve sayıda kural(lar) oluşturur.
- Sisteme standart yetkiler ile giren kullanıcı uzman tarafından oluşturulan soruları içeren anketi doldurur. Kullanıcının anket cevap anahtarı ile uzman tarafından oluşturulan kurallar işletilir ve yapılan analiz sonucunda kullanıcı için uygun olan veritabanındaki işlemciler listelenir.

Yazılımın genel çalışma şekli ve ekranları ile ilgili bilgi vermek gerekirse, yazılım açılmaya başladığında ilk olarak kullanıcıyı bilgilendirmek için şekil 2’de gösterilen geçiş ekranını kullanır (splash screen). Bu süre zarfında uygulama yönetici görev alır ve ilgili servisleri başlatır. Servislerin başarı ile başlatılması sonucunda şekil 3’de

gösterilen giriş ekranına geçilir ve kullanıcının sisteme giriş yapması için beklemeye geçilir.



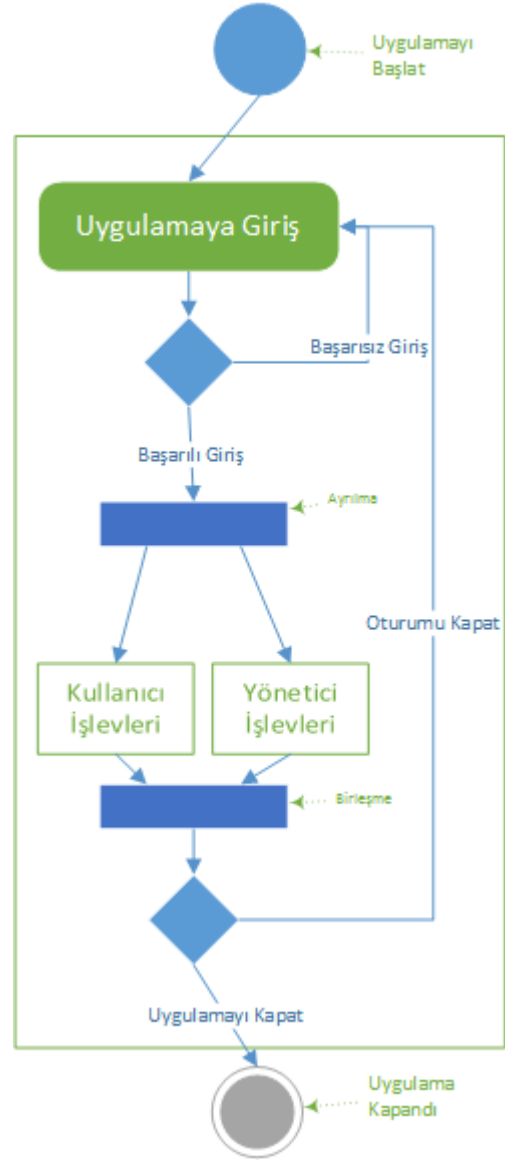
Şekil 2. Geçiş ekranı
(Splash Screen)



Şekil 3. Giriş ekranı
(Login screen)

Kullanıcı başarı ile giriş yaptığı durumda sahip olduğu yetkilendirmeye bağlı olarak yönetici ekranına veya standart kullanıcı ekranına yönlendirilir. Bu ekranlarda yapılabilen işlemler tasarım sürecinde belirlenmiş olup yetkilendirme düzeyi ile doğrudan ilişkilendirilmiştir.

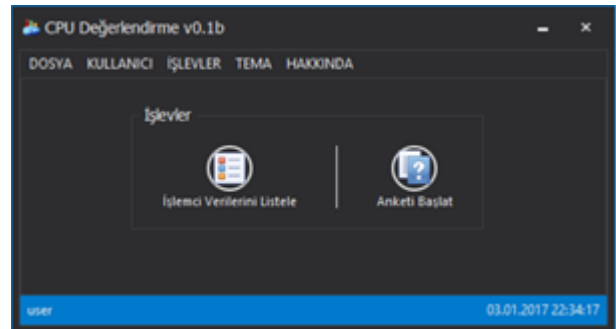
Kullanıcılar ilgili ekranlarında önceden belirlenmiş olan işlemleri gerçekleştirebilir. Bu işlemler ilerleyen başlıklarda detaylıca açıklanacaktır. Giriş yapılmış oturumlar oturumu kapatma özelliği ile kapatılarak giriş ekranına dönüş sağlanabilir veya doğrudan uygulama kapatma isteğinde bulunularak ilgili teyit işlemlerinin ardından uygulama kapatılabilir. Anlatılanları özetleyen yazılımın ana akış şeması aşağıda gösterilmiştir.



Şekil 4. Ana akış şeması
(Main flow chart)

3.2.1. Kullanıcı İşlevleri (User Functions)

Standart yetkiye sahip bir kullanıcı sisteme giriş yaptıığında şekil 5'de gösterilen kullanıcı ekranına yönlendirilir.

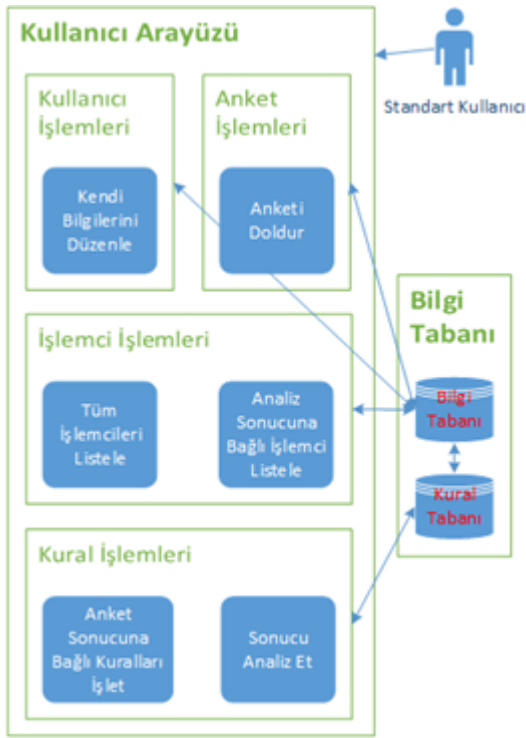


Şekil 5. Kullanıcı ana ekranı
(User main screen)

Bu ekran üzerinde kullanıcının yapabildiği işlevler aşağıda listelenmiştir.

- Kullanıcı bilgilerini düzenleyebilme. Yani kullanıcı adını ve parolasını değiştirebilme.
- Sistemde ekli işlemci bilgilerini listeleyebilme.
- Sistemde ekli soru ve seçenek bilgilerini listeleyebilme.
- Yazılım temasını değiştirebilme.
- Anket doldurarak yazılım temel işlevlerinden biri olan işlemci önerilerini almak. Bu özellik ilerleyen başlıklarda daha detaylı ele alınacaktır.

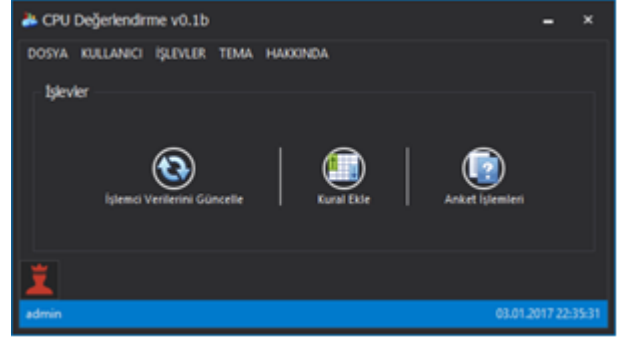
İşlev grupları ve bu işlevlerin bilgi tabanı ile olan etkileşimi şekil 6'da gösterilmiştir.



Şekil 6. Kullanıcı işlevleri
(User functions)

3.2.2. Yönetici İşlevleri (Admin Functions)

Yönetici yetkisine sahip bir kullanıcı sisteme giriş yaptığında şekil 7'de gösterilen kullanıcı ekranına yönlendirilir.

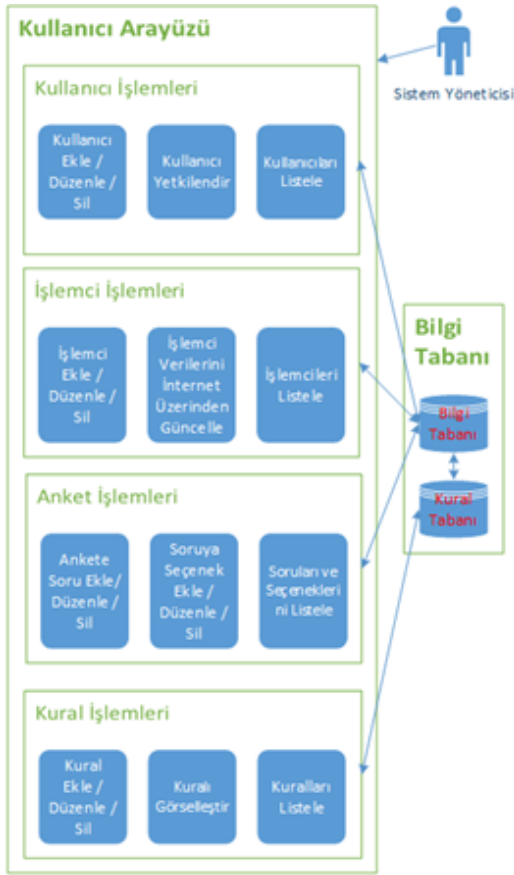


Şekil 7. Yönetici ana ekranı
(Admin main screen)

Bu ekran üzerinde yöneticinin yapabildiği işlevler aşağıda listelenmiştir.

- Sisteme yeni kullanıcı ekleyebilme.
- Kendi kullanıcı bilgilerini ve diğer kullanıcıların bilgilerini listeleyebilme, düzenleyebilme ve silebilme.
- Kullanıcılara yazılımı kullanma ihtiyaçlarına yönelik yönetici veya standart kullanıcı yetkilerinden birini verebilme.
- Güncel işlemci bilgilerini internet üzerinden çekebilme (internet bağlantısı gerekmektedir).
- İşlemci bilgilerini listeleyebilme, düzenleyebilme ve silebilme.
- Sisteme soru ve bu sorulara seçenekler ekleyebilme.
- Sistemdeki soru ve seçenekleri listeleyebilme, düzenleyebilme ve silebilme.
- Yazılım temasını değiştirebilme.
- Sisteme yeni kural ekleyebilme ve eklenen bu kuralı görselleştirebilme (ilerleyen başlıklarda anlatılacak).
- Kuralları listeleyebilme ve silebilme.

İşlev grupları ve bu işlevlerin bilgi tabanı ile olan etkileşimi şekil 8'de gösterilmiştir.



Şekil 8. Yönetici işlevleri
(Admin functions)

3.2.3. İşlemci İşlevleri (Processor Functions)

İşlemci piyasası sürekli gelişim ve değişim içerisinde olan, belli periyotlarda yeni işlemci ailelerinin var olanlarının yanına katıldığı aktif bir sektördür. Bu değişim içerisinde işlemci verileri kullanarak yapılacak olan bir analizin faydalı olabilmesi için verilerin hem nicelik hem de nitelik anlamında ihtiyacı karşılayabilmesini gerekmektedir.

Nicelik anlamındaki problemi çözmenin yolu, güncel işlemci verilerini yazılıma dahil etmekten geçmektedir. Nitelik anlamındaki problemi çözmeye yönelik ise piyasada bulunan işlemcilerden, son kullanıcılar için önem arz edenlerinin açıklanması ilk ihtiyaç olarak söylenebilir. Ayrıca işlemcilerin aynı karşılaştırma yazılımı (benchmark software) [19] kullanılarak elde ettiği skorları kullanmak veriyi nitelik anlamında daha da değerli hale getirmektedir.

Tüm bu ihtiyaçlar düşünüldüğünde yazılıma yetkili kullanıcı tarafından manuel işlemci verisi girme işlevinin yetersiz geleceği aşikardır. Bundan dolayı günlük olarak güncellenen bir işlemci karşılaştırma sitesinden güncel işlemci verilerini skorları ile beraber (yaklaşık 2500 işlemci verisi) ayrıştırarak (parsing) [20] yazılım veri sınıflarına ekleyen işlev gerçekleştirimi sağlanmıştır.

Yazılım tek bir işlev çağırısıyla 100'lerce işlemci verisini kendi veri sınıflarına çekip güncelleyebilmektedir.

Böylece analizin yapılacağı verinin nitelik ve nicelik anlamındaki problemlerinin önüne geçilmiştir. Şekil 9'da ilgili işlev çalıştırılarak internet üzerinden çekilen işlemci verilerini listeleyen ekran gösterilmiştir.

No	Açıklama	Performans - Fiyat	Kategori
1807	Intel Xeon ES-2679 v4 @ 2.50GHz	26236	\$1762
1808	Intel Xeon ES-2673 v4 @ 2.30GHz	24291	na
1809	Intel Xeon ES-2697 v4 @ 2.30GHz	22679	\$2787,01
1810	Intel Xeon ES-2696 v3 @ 2.30GHz	22743	\$2488
1811	Intel Xeon ES-2690 v4 @ 2.60GHz	22535	\$2090
1812	Intel Xeon ES-2699 v3 @ 2.30GHz	22528	\$3550
1813	Intel Xeon ES-2688 v4 @ 2.20GHz	21957	\$3228
1814	Intel Xeon ES-2696 v4 @ 2.20GHz	21954	na
1815	Intel Xeon ES-2699 v4 @ 2.20GHz	21951	\$4899,95
1816	Intel Xeon ES-2688 v3 @ 2.30GHz	21879	\$2890
1817	Intel Xeon ES-2697 v3 @ 2.60GHz	21852	\$2195
1818	Intel Xeon ES-2695 v3 @ 2.30GHz	20742	\$2913,31
1819	Intel Xeon ES-2687v v4 @ 3.00GHz	20458	na
1820	Intel Xeon ES-2690 v4 @ 2.40GHz	20078	na
1821	Intel Xeon ES-2695 v4 @ 2.10GHz	20043	\$2840,99
1822	Intel Core i7-6950X @ 3.00GHz	18978	\$1805
1823	Intel Xeon ES-2686 v3 @ 2.00GHz	18974	na
1824	Intel Xeon ES-2689 v4 @ 3.10GHz	19708	na
1825	Intel Xeon ES-2690 v3 @ 2.60GHz	19602	\$1909

Şekil 9. İşlemci listesi ekranı
(Processor list screen)

3.2.4. Soru ve Anket İşlevleri (Question and survey functions)

Yazılım kullanıcıların işlemci hakkındaki beklentilerini, kullanıcılara yönlendirmiş olduğu anket sorularını değerlendirerek vermektedir. Bu anket ise uzman tarafından hazırlanmış olan sorulardan oluşmaktadır. Böylelikle uzman bilgisi yazılıma aktarılmış olur [21].

Uzman, sisteme yönetici yetkisiyle girer ve hedeflediği mantıktaki soruları ve bu sorulara ait seçenekleri yazılıma ekler. Bu noktada yazılımı diğer uzman sistemlerden ayıran kısım sistemin tamamıyla dinamik içeriğe sahip olmasıdır. Yani kullanıcı istediği kadar soruyu sisteme dahil edebilir bu noktada bir üst limit yoktur. Yine aynı şekilde bir soruya istediği sayıda seçenek ekleyebilir. İster 2 şık ister 10 şık tüm bu dinamiklik kendine kural tabanında yer bulur. Yapının dinamikliği ile ilgili kısım bir sonraki başlık altında daha detaylı anlatılacaktır. Şekil 10'da dinamik soru ve seçenek ekleme ekranı gösterilmiştir.

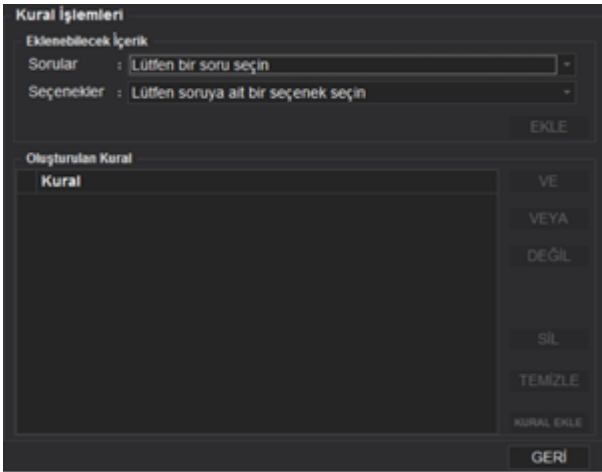
Şekil 10. Soru işlevleri ekranı
(Question functions screen)

3.2.5. Kural İşlevleri (Rule functions)

Yazılımda kullanılan kural işlevleri mevcut uzman sistemlerden farklı olarak tümüyle dinamik bir altyapıyla tasarlanmıştır. Uzman, uygulamaya istediği sayıda ve karmaşıklıkta kuralları ekleyebilir bunun için bir üst sınır ya da teknik bir engel söz konusu değildir.

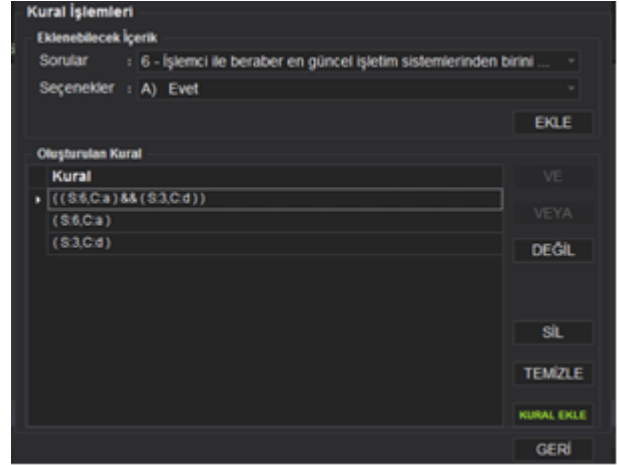
Yazılımda kural demek, daha önceden uzman tarafından uygulamaya eklenmiş olan bir ya da daha fazla sorunun şıklarının seçilme durumlarının koşullaştırılmış hali demektir. Daha açık bir ifade ile uzman “x” numaralı sonunun “y” harfli şıkkı üzerinden bir kural yazmak isterse, sistemde “S:x, C:y” şeklinde bir kural düğümü ekler. Bunun gibi bir ya da daha fazla kural düğümü kullanılarak bu düğümler üzerinde mantıksal cebir (boolean algebra) [22] işlevleri yapılır ve kural oluşturulur.

Kural eklemede kullanılan ekran piyasada gereksinim yönetimi için yoğunlukla kullanılan IBM Rational Doors yazılımı [23] örnek alınarak şekil 11’de gösterildiği gibi tasarlanmıştır.



Şekil 11. Kural işlevleri ekranı
(Rule functions screen)

Bu ekran kullanılarak uzman tarafından istenilen uzunlukta ve karmaşıklıkta kurallar oluşturulabilir. Kural içeriği için hiçbir üst sınır tanımlanmamıştır. Yeter ki uzman sisteme eklediği sorular ile mantıklı bir kural oluşturduğunu düşüncesine ulaşsın. Kural düğümleri üzerinde mantıksal cebir işlemlerinden !(değil), ||(veya), &&(ve) işlemleri yapılabilmektedir. Kurallarda oluşturulma sırasına bağlı önceliklendirme(parantezleme) yapılır. Böylelikle işlem öncelikleri de kural yapısında kendine yer bulmuş olur. Örnek bir kural ekleme senaryosu şekil 12’de gösterilmiştir.

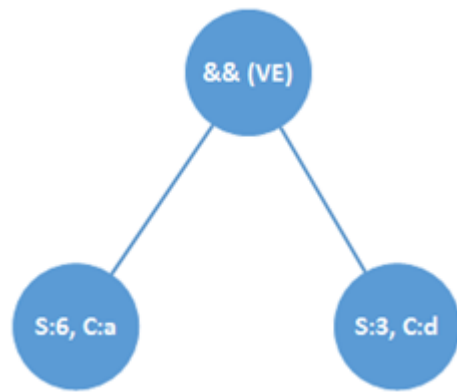


Şekil 12. Örnek kural ekleme senaryosu
(Sample rule adding scenario)

Şekil 12’deki örnekte “Soru:3, Cevap:d” kural düğümü ile “Soru:6, Cevap:a” kural düğümü sisteme eklenmiştir. Sonrasında bu iki kural düğümü &&(ve) mantıksal operatörüyle birleştirilmiş ve ortaya ((S:6, C:a) && (S:3, C:d)) şeklinde bir kural çıkmıştır. Bu kural daha açık bir ifade ile “Soru 6’da cevap olarak A şıkkı işaretlendiyse ve soru 3’de cevap olarak D şıkkı işaretlendiyse” anlamına gelmektedir ve kullanıcı anketi doldurduğunda bu kural işletilir geçerli ise 1(doğru) geçersiz ise 0(yanlış) olarak değerlendirilir.

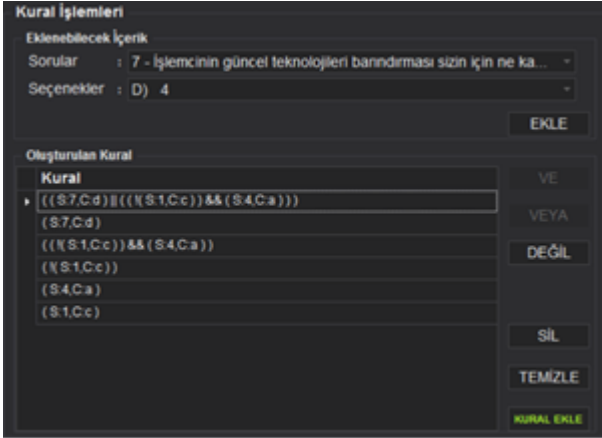
Yazılımda kurallar ağaç yapısı kullanılarak saklanmaktadır. Ağaç veri yapısı hem işlem önceliklerini yapısında tutabilmesi hem de sonsuz yapıda düğümü (node) barındırabilmesi özelliklerinden dolayı tercih edilmiştir [24].

Şekil 12’deki kural yazılım tarafında şekil 13’de gösterilen ağaç yapısı ile tutulur.



Şekil 13. Kural ağacı örneği
(Rule tree example)

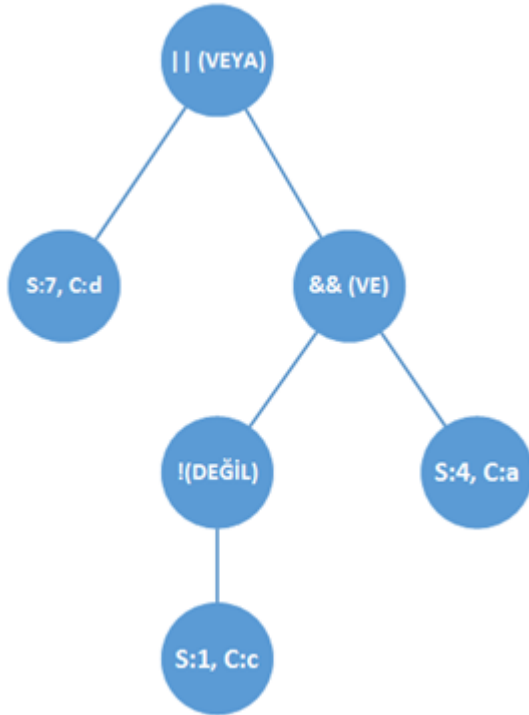
Daha karmaşık bir kural ekleme senaryosu şekil 14’de gösterilmiştir



Şekil 14. Örnek kural ekleme senaryosu 2
(Simple rule adding scenario 2)

Şekil 14'deki örnekte “Soru:1, Cevap:c”, “Soru:4, Cevap:a” ve “Soru:1, Cevap:c” kural düğümleri sisteme eklenmiş; sonrasında bu üç kural düğümü ile $((!(S:1, C:c) \&\& (S:4,C:a)))$ şeklinde bir kural oluşturulmuştur. Bu kural daha açık bir ifade ile “Soru 1’de cevap olarak C şıkkı işaretlenmediyse ve soru 4’de cevap olarak A şıkkı işaretlendiyse(öncelikli) veya soru 7’de D şıkkı işaretlendiyse” anlamına gelmektedir ve kullanıcı anketi doldurduğunda bu kural işletilir geçerli ise 1(doğru) geçersiz ise 0(yanlış) olarak değerlendirilir.

Şekil 14’deki kural yazılım tarafında şekil 15’de gösterilen ağaç yapısı ile tutulur.



Şekil 15. Kural ağacı örneği 2
(Rule tree example 2)

3.3. Uygulama Bilgi Tabanı (Application Knowledge Base)

Bilgi tabanı, bir konuda bir veya birden çok uzmanın bilgilerinin bir araya getirilmesiyle oluşur [25]. Yazılımda bilgi tabanı MsSQL 2014 veritabanı yönetim sistemi üzerinde tutulmuştur. Bilgi tabanı, veri tabanı ve kural tabanı olmak üzere iki bileşeni içermektedir.

Veritabanı: Problemin o andaki durumunu anlatan gerçekler ve belirli bir ana kadar elde edilmiş nitelik-değer çiftlerinden oluşur [26]. Uygulamada veritabanı işlemci ve soru verilerini tutmak için kullanılmıştır. Bununla beraber yazılımın çalışmasında gerekli görülen diğer veriler de veritabanına eklenmiştir.

Kural Tabanı: Uzman sistem, belirli durumlarda çalışan ve “eğer-o halde” veya diğer bir yapıda tanımlanan kurallar topluluğunu depolamak için bir belleğe sahip olmalıdır [25]. Kurallar, problemin ve sonuca ulaştıracak şartların düzgün şekilde ifade edildiği cümlelerdir. Uygulamada kural tabanı dinamik bir şekilde, “eğer-o halde” yapısına uygun tasarlanmıştır. Kuralların ağaç yapısında ve uzunluklarının değişken oluşu kural tabanına yazılan verilerin özel bir şablonda olmasını zorlamıştır. Kuralları, kural tabanına yazan ve kural tabanından okuyan özinelili (recursive) algoritmalar kullanılmış olup. Kural tabanı – Ağaç yapısı dönüşümlerini yapan adaptör işlevli metotlar tanımlanmıştır. Şekil 16’da yazılımda ağaç yapısında tutulan kurallarının, kural tabanında tutulma formatları gösterilmiştir.

KuralNo	DugunNo	UstDugunNo	DugunTur	SoruNo	SecenekNo	KuralGostereci	
1	1	0	-1	3	-1	-1	((!(S:1,C:c)) ((S:1,C:b)) ((S:1,C:a)))
2	2	1	0	2	-1	-1	NULL
3	2	0	-1	2	-1	-1	((!(S:2,C:e)) ((S:2,C:f)) ((S:2,C:d)))
4	2	1	0	0	2	4	NULL
5	2	2	0	0	2	2	NULL
6	2	3	0	0	2	3	NULL
7	3	0	-1	2	-1	-1	((!(S:3,C:e)) ((S:3,C:d)))
8	3	1	0	0	3	2	NULL
9	3	2	0	0	3	3	NULL
10	4	0	-1	3	-1	-1	((!(S:4,C:c)))
11	4	1	0	0	4	2	NULL
12	5	0	-1	2	-1	-1	((!(S:5,C:e)) ((S:5,C:b)))
13	5	1	0	0	5	2	NULL
14	5	2	0	0	5	1	NULL
15	6	0	-1	3	-1	-1	((!(S:6,C:b)))
16	6	1	0	0	6	1	NULL
17	7	0	-1	3	-1	-1	((!(S:7,C:e)) ((S:7,C:b)) ((S:7,C:a)))
18	7	1	0	2	-1	-1	NULL
19	8	0	-1	3	-1	-1	((!(S:8,C:b)))
20	8	1	0	0	8	1	NULL
21	9	0	-1	3	-1	-1	((!(S:9,C:b)))
22	9	1	0	0	9	1	NULL
23	10	0	-1	3	-1	-1	((!(S:10,C:e)))
24	10	1	0	0	10	0	NULL
25	11	0	-1	3	-1	-1	((!(S:11,C:b)))
26	11	1	0	0	11	1	NULL
27	12	0	-1	3	-1	-1	((!(S:12,C:e)) ((S:12,C:d)))
28	12	1	0	2	-1	-1	NULL

Şekil 16. Kural tabanı yapısı
(Rule database structure)

3.4. Uygulama Çıkarım Mekanizması (Application Inference Mechanism)

Çıkarım mekanizmasının görevi bilgi tabanını yorumlama ve kontroldür. Çıkarım mekanizması, kural tabanındaki kuralların kullanılarak anlamlı çıkarımların yapıldığı birimdir. Çıkarım mekanizması bilgi tabanındaki verileri ve kuralları kullanarak elde ettiği bu çıkarımları kullanıcıya aktarır. Çıkarım mekanizması kuralları yorumlamada, iki farklı arama metodu kullanır. Bunlar; ileri ve geriye zincirleme metotlarıdır [8,27]. Bu çalışmada, bilinen verilerden başlayarak uygun kuralın

bulunması durumunda ilgili kuralın mevcut şartlarını tatmin etmeyi temel alan "ileri zincirleme" metodu kullanılmıştır.

Uygulama, kullanıcının anketi doldurması ile kullanıcı anket cevaplarını analizi yapılmak üzere ilgili bileşene yönlendirir. Bu bileşen yazılımda ekli olan kuralları tamamını kullanıcı cevap anahtarına göre işletir ve ortaya kuralların ne kadarının başarılı ne kadarının başarısız olduğuna yönelik bir sonuç çıkarır.

Uygulamaya eklenen her bir kural, işlemci performansını ölçmeye yönelik olduğu için gerçekleşen kural sayısı arttıkça işlemci öneri ekranında listelenen işlemciler de aynı derecede karşılaştırma test sonucu yüksek olanlar arasından seçilir.

3.5. Uygulama Çıkarım Mekanizması (Application Inference Mechanism)

Açıklama ünitesi, elde edilen sonuçların kullanıcılara iletilmek üzere uzman sistem tarafından listelendiği bölümdür. Listenin doğruluğu geri bildirim için önemlidir [28-31]. Listelenen işlemciler tüm işlemci verilerinin bir alt kümesi olup, filtreleme işlemi işletilen kuralların doğruluk oranına göre yapılmaktadır.

4. SONUÇ VE ÖNERİLER (RESULTS AND SUCCESTION)

Cpu Değerlendirme Yazılımı, uzman bilgisi kullanarak işlemci seçiminde kullanıcıların doğru karar vermelerini hedefler. Bunu yaparken de güncel işlemci verilerini hem internet üzerinden hem de kullanıcıdan alabilir. Ayrıca uzman sistemlerin genelinden farklı olarak tamamıyla dinamik bir yapı kullanır böylece hem uzmana hem de kullanıcıya sunulacak anket ve analizinde ciddi bir esneklik sağlar.

Yazılımda performansa yönelik yapılan analizler geliştirilebilir. Kullanıcıya sunulan anket içeriği zenginleştirilerek, kullanıcıya önerilen işlemci listesi yalnızca performans tabanlı değil çeşitli anahtar kelimeler üzerinden yapılan filtrelemeler ile de genişletilebilir. Ayrıca işlemci tercihinde önemli bir kriter olan fiyat/performans parametresi internet üzerinden elde edilen bilgiler ışığında tüm işlemciler için elde edilebilir bir bilgi olmadığı için yazılımın sonuç analizinde devre dışı bırakılmıştır. Yazılımda internet üzerinden fiyat bilgisine ulaşılabilen işlemciler için bu analizi yapabilecek alt yapı tasarlanmış olup işlerlik kazandırılmamıştır.

KAYNAKLAR (REFERENCES)

[1] Pekkaya, Mehmet, and Mesut Aktogan. "Dizüstü Bilgisayar Seçimi: Dea, Topsis Ve Vikor İle Karşılaştırmalı Bir Analiz." AİBÜ-İİBF Ekonomik ve Sosyal Araştırmalar Dergisi, 10 (1), 107-126, 2014.

[2] Internet: U. Dal, <https://ugurdal.wordpress.com/tag/islemci-secimi/>, 21.01.2017.

[3] Internet: Hürriyet, <http://www.hurriyet.com.tr/intel-core-i5-mi-core-i7-mi-16396149>, 21.01.2017.

[4] M. Erkalın, M. H. Calp, İ. Şahin. "Çoklu Zekâ Kuramından Yararlanılarak Meslek Seçiminde Kullanılacak Bir Uzman Sistem Tasarımı ve Gerçekleştirilmesi.", International Journal of InformaticsTechnologies, 5-12, 2012.

[5] İ. Şahin, Uzman Sistem Kullanarak 2B'lu İzdüşümlerden Katı Model Oluşturma, Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2008.

[6] V.V. Nabiyev, Yapay Zeka, Seçkin Yayıncılık, Ankara, 445, 2005.

[7] M.H. Calp, N. Arıcı, Nesne Yönelimli Yazılım Kalitesi Açısından C&K Metrik Kümesini Değerlendiren Bir Uzman Modül Tasarımı Ve Uygulaması, DEÜ Mühendislik Fakültesi Mühendislik Bilimleri Dergisi, Cilt: 14 Sayı: 1 sh. 55-68 Ocak 2012.

[8] N. Allahverdi, Uzman Sistemler: Bir Yapay Zeka Uygulaması, Atlas Yayıncılık, İstanbul, 16-20, 2002.

[9] Y. S. Aydın, Visual Prolog İle Programlama (Yapay Zeka Ve Uzman Sistemler), Sistem Yayıncılık, İstanbul, 11-16, 2000.

[10] E. Turban, Decision Support and Expert Systems, Mac Millian Publishing Company, U.S.A., 1990.

[11] A. Koruvatan, Uzman Sistem Yardımı ile Araç Arızalarının Belirlenmesi, Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 21-24, 1999.

[12] Von Neumann, John, and John C. Oxtoby. John von neumann, American Mathematical Soc (1988).

[13] Hejlsberg, Anders, et al. C# Programming Language. Addison-Wesley Professional, 2010.

[14] Internet: Microsoft, <https://www.visualstudio.com/tfs/>, 21.01.2017.

[15]Internet: Devexpress, <https://www.devexpress.com/products/net/controls/winforms/>, 21.01.2017.

[16] Walters, Robert E., et al. "LINQ to SQL." Accelerated SQL Server 2008, 553-582, 2008.

[17] Liu, Xiaodong, Colin Combe, and Beihu Wang. "Component-based development.", U.S. Patent Application No. 11/786,311.

[18] Birrell, Andrew, An introduction to programming with C# threads. Technical report, Microsoft Corporation, May 2003. Manuscript available at <http://research.microsoft.com/~birrell/papers/ThreadsCSharp.pdf>, 2003.

[19] Internet: Passmark Software, <https://www.cpubenchmark.net/>, 21.01.2017.

[20] Gupta, Suhit, et al. "DOM-based content extraction of HTML documents." Proceedings of the 12th international conference on World Wide Web. ACM, 2003.

[21] İ. Kaya, Ş. Gözen, "Personel seçim sürecinde uzman sistem yaklaşımı ve Konya Büyükşehir Belediyesinde bir uygulama" Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 14, 355-376, 2005.

[22] S. Türker, "George Boole ve Mantık Cebiri." Kutadgubilig Felsefe-Bilim Araştırmaları Dergisi, 19, 283-314, 2011.

- [23] Internet: IBM, <http://www-03.ibm.com/software/products/tr/ratidoor>, 21.01.2017.
- [24] Samet, Hanan. The design and analysis of spatial data structures. Vol. 199. Reading, MA: Addison-Wesley, 1990.
- [25] H. Başak, İ. Şahin, M. Gülen, “İnsansız hava aracı kazalarının önlenmesi için uzman sisteme dayalı risk yönetim modeli, Teknoloji Dergisi, 11(3), 187-200, 2008.
- [26] İ. Şahin, H.R. Börklü, “2B görüşlerden otomatik katı modeller oluşturmada uzman bir yaklaşım”, Mühendislik Bilimleri Dergisi, 14(2), 111-123, 2008.
- [27] S. Üstkan, “Uzman Sistemler-Genel”, Yönlendirilmiş Çalışma, Sakarya Üniversitesi Adapazarı Meslek Yüksekokulu, Mart, 2007.
- [28] İ. Şahin, M.H. Calp, Ö. Akça, “Kredibilite Notu Değerlendirmeye Yönelik Bir Uzman Sistem Yaklaşımı”, Politeknik Dergisi, 14 (1), 79-83, 2011.
- [29] A. Öncü, Uzman Sistem Yaklaşımı İle Web Tabanlı Öğretim Değerlendirme Sisteminin Geliştirilmesi, Doktora Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2006.
- [30] Şahin, İ., Calp, M. H., & Özkan, A. (2014). An Expert System Design and Application for Hydroponics Greenhouse Systems. *Gazi University Journal of Science*, 27(2), 809-822.
- [31] Calp, M. H., & Şahin, İ. The determination by using fuzzy expert system of the usability level of website user interface design. *The Special Issue on Computer and Instructional Technologies*, 141.